-2-

<u>In the claims:</u>

For the Examiner's convenience, all pending claims are presented below with changes shown. Please cancel claims 10 and 17 without prejudice.

1.   (Currently Amended) A method comprising:

   receiving a string of data symbols; and

   compressing the string of data into a fixed sized compressed data block having a plurality of compressed symbols and dictionary elements, the <u>compressed</u> symbols and dictionary elements having a fixed length and a fixed offset within the <u>compressed data block</u>.

2.   (Original)   The method of claim 1 wherein compressing the data comprises:

   dividing a first symbol into a first component and a second component; and

   comparing the first component with the dictionary elements.

3.   (Original)   The method of claim 2 further comprising compressing the first component to form a first tag if the first component matches a dictionary element.

4.   (Currently Amended) The method of claim 3 wherein each <u>compressed</u> symbol· includes a tag to indicate a compression type.

5.   (Original)   The method of claim 3 further comprising storing the first component at a dictionary element if the first component does not match a dictionary element.

Docket No. 42P17035
Application No. 10/676,430   -2-

6.   (Original)   The method of claim 3 wherein compressing the data comprises:

dividing a second symbol into a second component and a second component; and

comparing the second component with the dictionary elements.

7.   (Currently Amended) A compression system:

a register to store a plurality of fixed length data symbols to be compressed;

compression logic to compress each of the plurality of data symbols to form a

compressed symbol, ~~the compressed symbols forming a compressed data block having a~~

~~fixed offset;~~ and

a plurality of dictionary registers to store dictionary elements, ~~having a fixed~~

~~length~~

<u>the compressed symbols and dictionary elements forming a compressed data</u>

<u>block having a fixed length and fixed offset</u>.

8.   (Original)   The system of claim 7 wherein each symbol is divided into a first

component and a second component.

9.   (Currently Amended) The method of claim 8 wherein the first and second

components are compressed into fixed length <u>compressed symbols</u> ~~tags~~.

10.   (Cancelled)

11.   (Original)   The system of claim 8 wherein the first component is received at

the compression logic and encoded to form a tag.

Docket No. 42P17035
Application No. 10/676,430   -3-

-4-

12.   (Original)   The system of claim 11 further comprising a buffer to store the tag and second component of each symbol as the compressed symbol.

13.   (Currently Amended) The system of claim 8 wherein the compression logic comprises:

dictionary matching logic to determine if the first component matches a dictionary element; and

~~constant~~ no match logic to determine if the second component has all ones or all zeroes.

14.   (Original)   The system of claim 13 wherein the compression logic comprises an encoder coupled to the match logic and the no match logic to encode the first component to form a tag if the first component matches a dictionary element, has all ones or zeroes.

15.   (Currently Amended) A method comprising:

receiving a fixed offset compressed data block having a plurality of dictionary elements and compressed symbols; and

decompressing each of the compressed symbols in parallel, by:

analyzing a tag component within a compressed symbol; and

decompressing the compressed symbol to form a symbol based upon the tag value.

Docket No. 42P17035
Application No. 10/676,430   -4-

-5-

16. (Original) The method of claim 15 wherein each of the compressed symbols are decompressed simultaneously.

17. (Cancelled)

18. (Currently Amended) The method of claim 15 17 wherein decompressing the compressed symbol to form a symbol based upon the tag value comprises:

decoding the tag to form a matched component of the symbol; and

combining the matched component with an unmatched component within the compressed symbol to form the symbol.

19. (Currently Amended) A decompression system comprising:

a plurality of decompression units to decompress a corresponding compressed symbol within a compressed data block having a plurality of compressed symbols and dictionary elements having a fixed length and fixed offset to generate an uncompressed symbol, wherein the decompression units decompress the compressed symbols in parallel.

20. (Original) The system of claim 19 wherein the compressed symbol comprises a tag component and an unmatched symbol component.

21. (Original) The system of claim 20 wherein each decompression unit comprises logic to decode the tag component of a compressed symbol to generate a matched symbol component.

Docket No. 42P17035
Application No. 10/676,430 -5-

-6-

22. (Original) The system of claim 21 wherein each decompression unit combines a matched symbol component with the unmatched symbol component to form an uncompressed symbol.

23. (Original) A computer system comprising:

a central processing unit (CPU);

a cache memory coupled to the CPU having a plurality of compressible cache lines to store additional data; and

a cache controller comprising compression logic to compress each of the plurality of cache lines by compressing the data within a compressed cache line into a fixed sized compressed data block having a plurality of offset compressed symbols and dictionary elements, the symbols and dictionary elements having a fixed length and fixed offset.

24. (Original) The computer system of claim 23 wherein the cache controller further comprises decompression logic to decompress compressed symbols within a compressed data block to generate uncompressed symbols.

25. (Original) The computer system of claim 24 wherein the decompression logic decompresses the compressed symbols in parallel.

26. (Original) A computer system comprising:

a central processing unit (CPU);

a cache memory coupled to the CPU having a plurality of compressible cache lines to store additional data;

Docket No. 42P17035
Application No. 10/676,430   -6-

a chipset, coupled to the CPU and the cache memory, including:

compression logic to compress each of the plurality of cache lines by
compressing the data within a compressed cache line into a fixed sized
compressed data block having a plurality of offset compressed symbols and
dictionary elements, the symbols and dictionary elements having a fixed length
and fixed offset; and

a main memory coupled to the chipset;

27.   (Original)   The computer system of claim 26 wherein the chipset further
comprises decompression logic to decompress compressed symbols within a compressed
data block to generate uncompressed symbols.

28.   (Original)   A method comprising:

receiving a fixed offset compressed data block having a plurality of dictionary
elements and compressed symbols; and

decompressing a randomly accessed and a first compressed symbol within the
compressed data block.

29.   (Original)   The method of claim 28 wherein decompressing the first
compressed symbol comprises:

analyzing a tag component within a compressed symbol; and

decompressing the compressed symbol to form a symbol based upon the tag
value.

Docket No. 42P17035
Application No. 10/676,430   -7-